

Listing of Claims:

1. **(Currently amended)** In a computer, a method of arranging a plurality of data members of an object class in a virtual memory system having units separately loadable from secondary memory into primary memory, the method comprising:

consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising a first group and a second group;

responsive to said consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising a first group and a second group, assigning memory locations for data members of the first group **of the object class** within a first unit of memory in the virtual memory system; and

responsive to said consulting metadata indicating how the data members of the object **class** are to be grouped into a plurality of separate groups comprising a first group and a second group, assigning memory locations for data members of the second group **of the object class** within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit.

2. **(Original)** The method of claim 1 wherein the plurality of separate groups comprises more than two groups, the method further comprising:

responsive to said consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising more than two groups, assigning memory locations for data members of groups other than the first group and the second group within respective units of memory in the virtual memory system separately loadable from the first unit of memory and the second unit of memory.

3. **(Original)** The method of claim 1 further comprising:

based on profile data of the object, generating the metadata to indicate more frequently referenced data members of the object are to be grouped in the first group and less frequently referenced data members are to be grouped in the second group.

4. (Original) The method of claim 1 further comprising:

based on profile data of the object, generating the metadata to indicate data members of the object with greater affinity are to be grouped in the first group and data members with lesser affinity are to be grouped in the second group.

5. (Original) The method of claim 4 further comprising:

generating the profile data by recording how many memory references were made to data members of the object during successive substantially-equal time periods;

wherein generating the metadata comprises calculating an affinity matrix from the profiling data.

6. (Original) In a computer, a method of arranging a plurality of data members of an object in a memory system, the method comprising:

identifying more frequently referenced data members out of the plurality of data members of the object by consulting profiling data for the object;

responsive to said identifying the more frequently referenced data members out of the plurality of data members of the object, grouping the more frequently referenced data members out of the plurality of data members of the object together into a group of more frequently referenced data members; and

responsive to said grouping the more frequently referenced data members into a group, assigning the data members from the group of more frequently referenced data members neighboring locations within the memory system.

7. (Original) The method of claim 6 wherein the memory system is a virtual memory system and the assigning assigns data members from the group of data members neighboring locations within the memory system by assigning data member from the group of data members a set of contiguous addresses in the virtual memory system.

8. (Original) The method of claim 6 wherein said identifying more frequently referenced data members out of the plurality of data members of the object by consulting profiling data for the object comprises:

ordering the data members according to the number of memory references made during profiling of the object.

9. (Original) The method of claim 6 wherein said identifying more frequently referenced data members out of the plurality of data members of the object by consulting profiling data for the object comprises:

ordering the data members according to affinity between the data members.

10. (Original) The method of claim 6 wherein said grouping the more frequently referenced data members out of the plurality of data members of the object together into a group of more frequently referenced data members comprises:

computing an affinity matrix for the data members;

transforming the affinity matrix into a cost matrix representing a graph of the data members;

constructing a minimum-cost spanning tree for the graph; and

traversing the minimum-cost spanning tree to determine members of the group.

11. (Original) The method of claim 6 further comprising:

generating the profiling data by counting how many memory references were made to data members of the object during successive method calls.

12. (Original) The method of claim 6 further comprising:

generating the profiling data by recording how many memory references were made to data members of the object during successive time periods.

13. (Original) The method of claim 6 further comprising generating the profiling data in a database by a method comprising:

maintaining counters indicative of how many memory references were made to data members of the object during successive observations;

adding the counters to an overall counter; and

writing a record to the database whenever the overall counter is likely to overflow based on an extrapolated forecast.

14. (Original) The method of claim 6 further comprising:

decorating a class definition of the object with metadata indicating a plurality of non-intersecting groups into which the data members have been placed.

15. (Original) The method of claim 6 wherein the neighboring locations within the memory system are within a unit of the memory system separately loadable from other units of the memory system.

16. (Original) The method of claim 15 wherein the unit of the memory system is a page separately loadable into primary memory.

17. (Original) The method of claim 6 further comprising:

identifying less frequently referenced data members out of the plurality of data members of the object by consulting the profiling data for the object;

responsive to said identifying the less frequently referenced data members out of the plurality of data members of the object, grouping the less frequently referenced data members out of the plurality of data members of the object together into a group of less frequently referenced data members; and

responsive to said grouping the less frequently referenced data members into a group, assigning the data members from the group of less frequently referenced data members at least one location separately-loadable from the locations of the more frequently referenced data members within the memory system.

18. **(Currently amended)** The method of claim 17 wherein the memory system is a virtual memory system having a primary memory and a secondary memory;
the virtual memory system comprises memory units separately-loadable into the primary memory;
said assigning the data members from the group of more frequently referenced data members neighboring locations within the memory system comprises assigning the more frequently referenced data members locations within a first set of one or more memory units in the virtual memory system; and
said grouping the less frequently referenced data members out of the plurality of data members of the object together into a group of more frequently referenced data members comprises assigning the less frequently referenced data members locations within a second set of one or more memory units in the virtual memory system; and
the intersection of the first set of memory units and the second set of memory units is ~~the~~ a null set.

19. (Original) The method of claim 18 wherein the memory units are memory pages.
20. (Original) The method of claim 18 wherein the memory units are of variable size.

21. (Original) In a computer, a method of arranging a plurality of data members of an object in a memory system by separating the plurality of data members, wherein the memory system comprises a primary and a secondary memory and the memory system comprises a set of memory units separately loadable into the primary memory, the method comprising:

identifying more frequently referenced data members out of the plurality of data members of the object and less frequently referenced data members out of the plurality of data members of the object by consulting profiling data for the object;

responsive to said identifying the more frequently referenced data members out of the plurality of data members of the object and less frequently referenced data members out of the plurality of data members of the object, grouping the more frequently referenced data members out of the plurality of data members of the object together into first group of data members and the less frequently referenced data members out of the plurality of data members of the object together into a second group of data members; and

responsive to said grouping, assigning the data members from the first group of data members locations within a first set of memory units in the memory system and the data members from the second group of data members locations within a second set of memory units in the memory system separate from the first set.

22. (Original) The method of claim 21 wherein said grouping the more frequently referenced data members out of the plurality of data members of the object together into first group of data members and the less frequently referenced data members out of the plurality of data members of the object together into a second group of data members comprises the following:

ordering the data members by how many times referenced in memory during profiling;
and

dividing the data members into the first group and the second group based on a threshold.

23. (Original) The method of claim 22 further comprising:

calculating an average number of references observed during profiling; wherein the threshold is the average number of references.

24. (Original) The method of claim 22 further comprising:
calculating an average number of references observed during profiling;
receiving a specified number representing a desired multiple of an average; wherein the threshold is the desired multiple times the average number of references.

25. (Original) The method of claim 22 further comprising:
receiving a specified number of memory references indicative of a desired threshold number of memory references observed during profiling; wherein the threshold is the desired threshold number of memory references.

26. (Original) The method of claim 22 further comprising:
calculating a standard deviation for a number of references observed during profiling;
receiving a specified number representing a desired multiple of the standard deviation;
wherein the threshold is the desired multiple times the standard deviation.

27. (**Currently amended**) In a computer, a method of splitting data members of an object class into a set of groups to be placed at separately-loadable units of memory in a memory system by consulting profile data indicating how many times the data members were referenced in memory, the method comprising:

assembling a list of the data members indicating how many times each data member in the list was referenced in memory;

selecting a threshold value to be applied to the list of data members according to a threshold framework;

applying the threshold value to the list of data members to identify a candidate cold group;

determining whether to split the data members based whether the candidate cold group is acceptable to a policy framework; and

responsive to said determining, selecting between splitting the data members into a set of groups comprising a hot group and a cold group based on the threshold **framework** having affirmatively determined to split the data members and omitting to split the data members into a set of groups having determined not to split the data members.

28. (Original) The method of claim 27 wherein the threshold framework supports options comprising:

specifying a threshold value in terms of absolute number of memory references to data members of the object class;

specifying a threshold value in terms of multiples of an average number of memory references to data members of the object class; and

specifying a threshold value in terms of standard deviations from an average number memory referenced to data members of the object class.

29. (Original) The method of claim 27 wherein the threshold framework supports specification of a default threshold value to be applied to a plurality of classes and specification of an override threshold value for a particular class.

30. (Original) The method of claim 27 wherein the policy framework supports options based on a size of the candidate cold group and overhead is introduced into the cold group and a group referring to the cold group when the data members are split, the options comprising:

determining whether to split the data members based on whether the size of the candidate cold group is greater than the overhead introduced into the referring group;

determining whether to split the data members based on whether the size of the cold group is greater than a sum of the overhead introduced into the referring group and the overhead introduced into the cold group;

determining whether to split the data members based on whether the size of the cold group is greater than a specified fraction of a sum of the overhead introduced into the referring group and the overhead introduced into the cold group; and

determining whether to split the data members based on whether the size of the cold group is greater than a sum of the overhead introduced into the referring group and the overhead introduced into the cold group and whether the cold group is accessed more frequently than an average number of memory references.

31. **(Currently amended)** A computer-implemented method for optimizing the data layout of software comprising a set of object class definitions, the method comprising:

- instrumenting the software to generate profile data indicating how frequently data members of the object class definitions are referenced in memory;
- grouping the data members of the object class definitions into groups according to how frequently the data members of the object class definitions are referenced in memory, wherein the groups comprise a more frequently referenced group and a less frequently referenced group;
- associating metadata indicative of the groups with the set of object class definitions; and
- at runtime, consulting the metadata indicative of the groups to determine how to arrange the data members within memory, placing the data members in the more frequently referenced group in a first unit of memory and the data members in the less frequently referenced group in a second unit of memory separately loadable from the first unit of memory into primary memory.

[[,]]

32. (Original) The method of claim 31 further comprising:

- declaratively specifying in the software that a specified object class definition is not to be optimized; and
- during said grouping, inhibiting grouping of data members of the specified object class definition.

33. (Original) The method of claim 31 further comprising:

- at runtime, loading data members in the first group into primary memory; and inhibiting loading data members in the second group into primary memory.

34. **(Currently amended)** A method of specifying data layout optimization for software comprising a set of objects belonging to object classes, the method comprising:

- profiling the software to generate profile data indicating how many times each data member of each of the set of objects was referenced in memory during the profiling;
- specifying a default threshold selected from the group consisting of a linear threshold, an absolute threshold, and a Gaussian threshold;
- for at least one object class, specifying a threshold selected from the group consisting of a linear threshold, an absolute threshold, and a Gaussian threshold, wherein the threshold of the at least one object class is other than the default threshold; and
- splitting the data members of the objects into at least two groups based according to the thresholds, wherein each of the at least two groups is to reside in a unit of memory separately loadable in a virtual memory system.

35. **(Original)** The method of claim 34 wherein said splitting is selectively performed responsive to consulting the threshold;

- and said splitting is inhibited if all data members of an object class fall above the threshold.

36. (Original) A computer-implemented method for optimizing the layout of data members of an object class within a memory system comprising primary memory and secondary memory, the method comprising:

- generating profiling data by recording how frequently memory references are made to data members defined for the object class during profiling;

- separating the data members into a hot group and a cold group based on how frequently memory references are made to the data members defined for the object during profiling;

- at runtime, assigning data members in the hot group locations in a first unit of memory of the memory system and assigning data members in the cold group locations in a second unit of memory of the memory system, wherein the first unit of memory can be loaded into primary memory without loading the second unit of memory; and

- at runtime, loading the first unit of memory into primary memory and not loading the second unit of memory into primary memory.

37. (Original) A computer-implemented method for optimizing the layout of data members of an object class within a memory system comprising primary memory and secondary memory, the method comprising:

- generating profiling data by recording how many memory references are made to data members defined for the object class at successive time intervals;

- generating an affinity matrix indicating affinities between the data members defined for the object;

- transforming the affinity matrix into a minimum-cost spanning tree, wherein the data members defined for the object are vertices in the minimum-cost spanning tree;

- traversing the minimum-cost spanning tree to visit each data member in a connected graph and recording the order of visiting each data member;

- dividing the data members into a hot group and a cold group based on a threshold edge cost;

- recording said dividing in metadata associated with a definition of the object class; and

- at runtime, consulting the metadata to place data members of the hot group into a unit of memory separately loadable into primary memory from a unit of memory into which the cold group are placed.

38. (Original) The method of claim 37 wherein one or more data members appear in the minimum-cost spanning tree in a graph disconnected from the connected graph, the method further comprising:

responsive to determining the data members appear in a graph disconnected from the connected graph, placing data members in the disconnected graph in a group separate from the hot group.

39. (Original) In a computer, a method of monitoring affinity among a plurality of data members of an object class in a memory system, the method comprising:

constructing a table indicating how many times data members of the object were referenced during each of a plurality of time periods during profiling;

computing an affinity matrix for the object class with a dot product operation, wherein the affinity matrix indicates an affinity between a first data member of the object and a second data member of the object; and

displaying the affinity matrix as a graph for consideration by a user.

40. (Original) The method of claim 39 further comprising:

generating a database record by tracking how many time data members of the object are referenced during successive time periods by setting an expiring timer; and

writing the database record to a database of profiling data upon expiration of the timer.

41. (Original) In a computer having a virtual memory system, a system for performing data layout optimization on software comprising a set of object class definitions, the system comprising:

profile data for the object class definitions indicating how many times data members of objects instantiated according to the object class definitions were referenced in memory during profiling;

a data member grouper operative to consult the profile data and group the data members of an object into at least two groups, wherein the groups comprise a more frequently referenced group of data members and a less frequently referenced group of data members, wherein the data member grouper is further operative to annotate the object class definitions to indicate into which of the groups data members defined in the object class definitions fall.

42. (Currently amended) **An One or more computer-readable media having computer-executable instructions implementing an** execution engine operable to perform operations on an object of an object class defined by an object class definition and residing in a virtual memory system having units separately loadable into primary memory, the execution engine comprising:

means for receiving a request from software to load the object of the object class;

means for maintaining information about a plurality of data members of the object class, wherein the information explicitly indicates the plurality of data members are to be placed on different plural units within the virtual memory system; and

means for arranging the plurality of data members of the object of the object class within plural units of memory in the virtual memory system according to the information about the plurality of data members of the object class, wherein the means for arranging is operative to consult the information indicating information about the data members responsive to the request to arrange.

43. (Currently amended) The ~~method~~ one or more computer-readable media of claim 42 wherein said means for arranging is a class loader operable to receive a layout directive specifying an explicit layout, a sequential layout, or an automatic layout.

44. (Currently amended) The ~~execution engine~~ one or more computer-readable media of claim 42 wherein the units of the virtual memory system are pages of a fixed size.

45. (Currently amended) The ~~execution engine~~ one or more computer-readable media of claim 42 further comprising:

means for collecting profiling information indicating memory references to data members during execution of the software;

wherein said means for arranging is operable to consult the profiling information indicating memory references to data members during execution of the software to determine how to arrange the plurality of data members.

46. (Original) In a computer-readable medium, a data structure for indicating how a plurality of data members defined by an object class definition of an object class are to be arranged in a memory system at runtime of an object, wherein the object is an instance of the object class, wherein the memory system supports separately-loadable units, the data structure comprising:

for the object class, a layout directive field indicating whether layout of the data members of the object class is to be performed according to a scheme specified in the data structure; and

for at least two data members of the object class, a grouping field indicating a group into which data members are to be placed at runtime, wherein the grouping field for a first at least one data member of the object class indicates the at first at least one data member of the object class is to be placed into a first group residing in a first unit of the memory system at runtime, and the grouping field for a second at least one data member of the object class indicates the at second at least one data member of the object class is to be placed into a second group residing in a second unit of the memory system separately-loadable from the first unit at runtime.

47. **(Currently amended)** The data structure of claim 46 in a set of data structures for indicating data layout, wherein the object class is a first object class and the data structure comprises an entry for a second object class, wherein the second object class comprises a ~~entries~~ the layout directive field ~~is~~ defined to accommodate values for an automatic layout, an explicit layout, and a sequential layout.

48. (Original) The data structure of claim 46 wherein the grouping field groups data members of the class according to their affinity as determined by calculating an affinity matrix for the data members and transforming the affinity matrix into a minimum-cost spanning tree.

49. (Original) The data structure of claim 46 wherein the grouping field groups data members of the class according to how many times the data members were referenced in memory during profiling.

50. (Original) In a computer-readable medium, a set of data structures for a respective set of object classes, wherein a data structure out of the set indicates how data members of the respective object class out of the set are to be arranged in a memory system at runtime of an instance of an object of the respective object class, wherein the memory system supports separately-loadable units, the data structure for the respective object class comprising:

a layout directive field indicating whether to use an automatic layout, an explicit layout, or a sequential layout when arranging the data members of the respective object class at run time;
grouping fields for the data members of the respective object class, the grouping fields indicating how data members of the respective object are to be divided into the separately-loadable units of the memory system at runtime.

51. (Original) The set of data structures of claim 50 wherein the data structure for the respective object class further comprises:

offset fields for the data members of the respective object class indicating at what position within a separately-loadable unit of the memory system the data members are to be placed.

52. (Original) In a computer, a method of laying out data members of an object having a plurality of data members, the method comprising:

collecting profiling data for the object, wherein the profiling data indicates more frequently and less frequently accessed data members of the object;

recording the more frequently and less frequently accessed data member of the object in metadata associated with a definition of the object;

responsive to consulting the metadata, arranging the data members of the object into a plurality of separately loadable blocks to group at least one more frequently accessed member into a first one of the plurality of separately loadable blocks and at least one less frequently accessed data member into a second one of the plurality of separately loadable blocks.

53. (Currently amended) In a computer system having a memory system, a method of dividing an object comprising a plurality of data members into plural units of memory in the memory system, the method comprising:

observing execution of the object to determine a set of more referenced data members out of the plurality of data members and a set of lesser referenced data members out of the plurality of data members;

responsive to said observing, loading the more referenced data members into a first unit of memory in the memory system; and

responsive to said observing, loading the lesser reference data members into a second unit of memory in the memory system, wherein the second unit of memory in the memory system is loadable in the memory system separately from the first unit of memory in the memory system.